Claim Amendments

Claim 1 (currently amended): A method for creating an appearance of texture in a computer image comprising the steps of:

inputting a point [[$\{x_d\}$]] (x, y, z) in D-dimensional geometric space R^3 described via six eight-bit quantities i, j, k, u, v, w, where i, j, k are the greatest integers not greater than x, y, z, respectively, and u, v, w signify a fractional position of x, y, z above i, j, k to eight-bit precision~~D M bit quantities i_d and D N bit quantities u_d, where i_d are M bit representations of greatest integers not > x_d and u_d are N bit representations of remainders (x_d-i_d), where M and N are integers ≥ 4 and D=3~~, in a computer;

computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into an upper half of the table and entries of the upper half of the table index into the lower half of the table;

computing a contribution from each vertex using the hash-value; and

combining with the computer the contribution from each vertex into a single interpolated result so that each 3 dimensional evaluation of one x, y, z triplet requires only one pipelined clock cycle~~, where one clock cycle is between 200-300 MHz~~.

Claim 2 (original): A method as described in Claim 1 wherein the computing a hash value step includes computing eight five bit pseudo-random hash values hn, one hash value for each of the eight vertices of the surrounding unit cube C using six + modules and seven L modules.

Claim 3 (original): A method as described in Claim 2 wherein the computing a contribution step includes computing for each vertex of the surrounding unit cube C the contribution of each vertex with three + modules and eight H modules.

Claim 4 (original): A method as described in Claim 3 wherein the combining step includes the step of combining the contribution from each vertex into a single result using 3 ease-curve s modules.

Claim 5 (original): A method as described in Claim 4 wherein the computing a hash value step includes the step of implementing each L module as a look-up table which simultaneously retrieves 2 successive table entries, the table has n/2 rows with 2 data bits per

row, where top B-1 controlled bits are used to reflect a row r, and where a lowest control bit latches between selecting entry r and r + 1 for lowest b bits, and swapping lower b bits with upper b bits at a point where related data exits the table.

Claim 6 (previously presented): A method as described in Claim 5 wherein the computing the contribution step includes the steps of subtracting 28 from each u, v, w, computing a gradient direction from each hash value hn, performing an inner product between the gradient direction and the associated fractional position from the associated vertex.

Claim 7 (original): A method as described in Claim 6 wherein the computing the gradient direction includes the step of mapping a lower 6 bits from a last stage of the L modules into a fixed set of gradient directions such that a length of each component of every sector is a power of 2 which allows the inner product to be done using no multiples, only adds and shifts.

Claim 8 (original): A method as described in Claim 7 wherein the mapping step includes the step of choosing the gradients so as to be symmetrical about the principal axis, the edge diagonals and the corner diagonals of the surrounding unit cube C.

Claim 9 (original): A method as described in Claim 8 wherein the combining step includes the step of using 7 linear-interrelation modules L to perform a trilinear interpolations from the eight vertices of C using the 3 ease curves as interpolants.

Claim 10 (original): A method as described in Claim 9 wherein the combining step includes the step of computing each ease curve in each dimension using a pre-computed entry table S sampling at intervals of 2-7 from a piecewise second order polynomial: if ($t < ½$) then ($2t^2$) else ($-2t^2+4t-1$).

Claim 11 (previously presented): A method as described in Claim 10 wherein the using step includes the step of using the seven linear interpolations modules I, arranged into three successive stages, wherein a first stage of the three stages eight values are reduced to four values, interpolating in x; the second stage of the four values are reduced to two, interpolating in y; and the third stage, the two values are reduced to one, interpolating in z.

Claim 12 (currently amended): An apparatus for creating an appearance of texture in a computer image comprising:

a computer;

a mechanism for inputting a point [[{x_d}]] (x, y, z) in D-dimensional geometric space R^3 described via six eight-bit quantities i, j, k, u, v, w, where i, j, k are the greatest integers not greater than x, y, z, respectively, and u, v, w signify a fractional position of x, y, z above i, j, k to eight-bit precision ~~D M bit quantities i_d and D N bit quantities u_d, where i_d are M bit representations of greatest integers not > x_d and u_d are N bit representations of remainders (x_d-i_d), where M and N are integers ≥ 4 and D=3~~, in the computer;

a mechanism for computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into an upper half of the table and entries of the upper half of the table index into the lower half of the table;

a mechanism for computing a contribution from each vertex using the hash-value; and

a mechanism for combining with the computer the contribution from each vertex into a single interpolated result so that each 3 dimensional evaluation of one x, y, z triplet requires only one pipelined clock cycle~~, where one clock cycle is between 200-300 MHz~~.